

# Steiner Transitive-Closure Spanners of $d$ -Dimensional Posets

Piotr Berman\*   Arnab Bhattacharyya†   Elena Grigorescu‡   Sofya Raskhodnikova\*  
 David P. Woodruff§   Grigory Yaroslavl'tsev\*

November 30, 2010

## Abstract

Given a directed graph  $G = (V, E)$  and an integer  $k \geq 1$ , a  $k$ -transitive-closure-spanner ( $k$ -TC-spanner) of  $G$  is a directed graph  $H = (V, E_H)$  that has (1) the same transitive-closure as  $G$  and (2) diameter at most  $k$ . In some applications, the shortcut paths added to the graph in order to obtain small diameter can use Steiner vertices, that is, vertices not in the original graph  $G$ . The resulting spanner is called a *Steiner transitive-closure spanner* (Steiner TC-spanner).

Motivated by applications to property reconstruction and access control hierarchies, we concentrate on Steiner TC-spanners of directed acyclic graphs or, equivalently, partially ordered sets. In these applications, the goal is to find a sparsest Steiner  $k$ -TC-spanner of a poset  $G$  for a given  $k$  and  $G$ . The focus of this paper is the relationship between the dimension of a poset and the size of its sparsest Steiner TC-spanner. The dimension of a poset  $G$  is the smallest  $d$  such that  $G$  can be embedded into a  $d$ -dimensional directed hypergrid via an order-preserving embedding.

We present a nearly tight lower bound on the size of Steiner 2-TC-spanners of  $d$ -dimensional directed hypergrids. It implies better lower bounds on the complexity of local reconstructors of monotone functions and functions with low Lipschitz constant. The proof of the lower bound constructs a dual solution to a linear programming relaxation of the Steiner 2-TC-spanner problem. We also show that one can efficiently construct a Steiner 2-TC-spanner, of size matching the lower bound, for any low-dimensional poset. Finally, we present a lower bound on the size of Steiner  $k$ -TC-spanners of  $d$ -dimensional posets that shows that the best-known construction, due to De Santis *et al.*, cannot be improved significantly.

---

\*Pennsylvania State University, USA. {berman, sofya, grigory}@cse.psu.edu. S.R. and G.Y. are supported by NSF / CCF CAREER award 0845701. G.Y. is also supported by University Graduate Fellowship and College of Engineering Fellowship.

†Massachusetts Institute of Technology, USA. abhatt@mit.edu

‡Georgia Institute of Technology, USA. elena@cc.gatech.edu. Supported in part by NSF award CCR-0829672 and NSF award 1019343 to the Computing Research Association for the Computing Innovation Fellowship Program.

§IBM Almaden Research Center, USA. dpwoodru@us.ibm.com.

# 1 Introduction

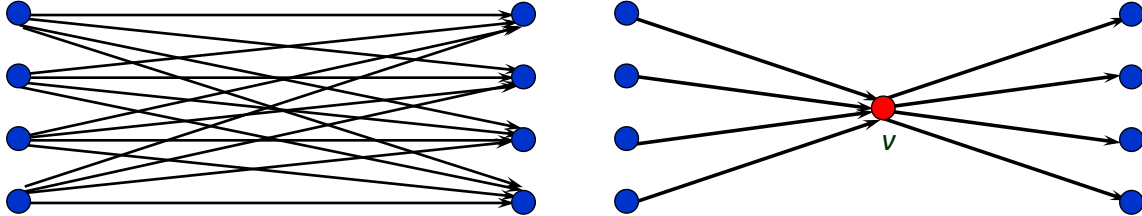
Graph spanners were introduced in the context of distributed computing [18], and since then have found numerous applications. Our focus is on transitive-closure spanners, introduced explicitly in [8], but studied prior to that in many different contexts [11, 10, 26, 3, 12, 22, 9, 23, 24, 13, 16, 6, 5, 4].

Given a directed graph  $G = (V, E)$  and an integer  $k \geq 1$ , a  **$k$ -transitive-closure-spanner** ( $k$ -TC-spanner) of  $G$  is a directed graph  $H = (V, E_H)$  satisfying: (1)  $E_H$  is a subset of the edges in the transitive closure of  $G$ ; (2) for all vertices  $u, v \in V$ , if  $d_G(u, v) < \infty$  then  $d_H(u, v) \leq k$ . That is, a  $k$ -TC-spanner is a graph with a small diameter that preserves the connectivity of the original graph. The edges from the transitive closure of  $G$  that are added to  $G$  to obtain a TC-spanner are called *shortcut edges* and the parameter  $k$  is called the *stretch*.

TC-spanners have numerous applications, and there has been lots of work on finding sparse TC-spanners for specific graph families. (See [19] for a survey.) In some applications of TC-spanners (in particular, to access control hierarchies [5, 6, 21, 4]), the shortcuts can use *Steiner* vertices, that is, vertices not in the original graph  $G$ . The resulting spanner is called a *Steiner TC-spanner*.

**Definition 1.1** (Steiner TC-spanner). *Given a directed graph  $G = (V, E)$  and an integer  $k \geq 1$ , a **Steiner  $k$ -transitive-closure-spanner** (**Steiner  $k$ -TC-spanner**) of  $G$  is a directed graph  $H = (V_H, E_H)$  satisfying: (1)  $V \subseteq V_H$ ; (2) for all vertices  $u, v \in V$ , if  $d_G(u, v) < \infty$  then  $d_H(u, v) \leq k$  and if  $d_G(u, v) = \infty$  then  $d_H(u, v) = \infty$ . Vertices in  $V_H \setminus V$  are called Steiner vertices.*

For some graphs, Steiner TC-spanners can be significantly sparser than ordinary TC-spanners. For example, consider a complete bipartite graph  $K_{\frac{n}{2}, \frac{n}{2}}$  with  $n/2$  vertices in each part and all edges directed from the first part to the second. Every ordinary 2-TC-spanner of this graph has  $\Omega(n^2)$  edges. However,  $K_{\frac{n}{2}, \frac{n}{2}}$  has a Steiner 2-TC-spanner with  $n$  edges: it is enough to add one Steiner vertex  $v$ , edges to  $v$  from all nodes in the left part, and edges from  $v$  to all nodes in the right part. Thus, for  $K_{\frac{n}{2}, \frac{n}{2}}$  there is a linear gap between the size of the sparsest Steiner 2-TC-spanner and the size of an ordinary 2-TC-spanner.



We concentrate on Steiner TC-spanners of directed *acyclic* graphs (DAGs) or, equivalently, partially ordered sets (posets) because they represent the most interesting case in applications of TC-spanners. In addition, there is a reduction from constructing TC-spanners of graphs with cycles to constructing TC-spanners of DAGs, with a small loss in stretch ([19], Lemma 3.2), which also applies to Steiner TC-spanners.

The goal of this work is to understand the minimum number of edges needed to form a Steiner  $k$ -TC-spanner of a given graph  $G$  as a function of  $n$ , the number of nodes in  $G$ . More specifically, motivated by applications to access control hierarchies [5, 6, 21, 4] and property reconstruction [7, 17], described in Section 1.2, we study the relationship between the dimension of a poset and the size of its sparsest Steiner TC-spanner. The *dimension* of a poset  $G$  is the smallest  $d$  such that  $G$  can be embedded into a  $d$ -dimensional directed hypergrid via an order-preserving embedding. (See Definition 2.1). Atallah *et al.* [4], followed by De Santis *et al.* [21], use Steiner TC-spanners in key management schemes for access control hierarchies. They argue that many access control hierarchies are low-dimensional posets that come equipped with an embedding demonstrating low dimensionality. For this reason, we focus on the setting where the dimension  $d$  is small relative to the number of nodes  $n$ .

We also study the size of sparsest (Steiner) 2-TC-spanners of specific posets of dimension  $d$ , namely,  $d$ -dimensional directed hypergrids. Our lower bound on this quantity improves the result in [7] and nearly matches the upper bound from that paper. It implies that our construction of Steiner 2-TC-spanners of  $d$ -dimensional posets cannot be improved significantly. It also has direct implications for property reconstruction. The focus on stretch  $k = 2$  is motivated by both applications.

## 1.1 Our Results

**Steiner 2-TC-spanners of directed  $d$ -dimensional grids.** The *directed hypergrid*, denoted  $\mathcal{H}_{m,d}$ , has vertex set<sup>1</sup>  $[m]^d$  and edge set  $\{(x, y) : \exists \text{ unique } i \in [d] \text{ such that } y_i - x_i = 1 \text{ and if } j \neq i, y_j = x_j\}$ . We observe (in Corollary 2.4) that for the grid  $\mathcal{H}_{m,d}$ , Steiner vertices do not help to create sparser  $k$ -TC-spanners. In [7], it was shown that for  $m \geq 3$ , sparsest (ordinary) 2-TC-spanners of  $\mathcal{H}_{m,d}$  have size at most  $m^d \log^d m$  and at least  $\Omega\left(\frac{m^d \log^d m}{(2d \log \log m)^{d-1}}\right)$ . They also give tight upper and lower bounds for the case of constant  $m$  and large  $d$ . Our first result is an improvement on the lower bound for the hypergrid for the case when  $m$  is significantly larger than  $d$ .

**Theorem 1.1.** *Every (Steiner) 2-TC-spanner of  $\mathcal{H}_{m,d}$  has  $\Omega\left(\frac{m^d (\ln m - 1)^d}{(4\pi)^d}\right)$  edges.*

The proof of Theorem 1.1 constructs a dual solution to a linear programming relaxation of the Steiner 2-TC-spanner problem. We consider an integer linear program for the sparsest 2-TC-spanner of  $\mathcal{H}_{m,d}$ . Our program is a special case of a more general linear program for the sparsest directed  $k$ -spanner of an arbitrary graph  $G$ , used in [8] to obtain an approximation algorithm for that problem. As explained in [8], the general program has an integrality gap of  $\Omega(n)$ . However, we show that for our special case the integrality gap is small and, in particular, does not depend on  $n$ . Specifically, we find a solution to the dual linear program by selecting initial values that have a combinatorial interpretation: they are expressed in terms of the *volume* of  $d$ -dimensional *boxes* contained in  $\mathcal{H}_{m,d}$ . For example, the dual variable corresponding to the constraint that enforces the existence of a length-2 path from  $u$  to  $v$  in the 2-TC-spanner is initially assigned a value inversely proportional to the number of nodes on the paths from  $u$  to  $v$ . The final sum of the constraints is bounded by an integral which, in turn, is bounded by an expression depending only on the dimension  $d$ .

We note that the best lower bound known previously [7] was proved by a long and sophisticated combinatorial argument that carefully balanced the number of edges that stay within different parts of the hypergrid and the number of edges that cross from one part to another. Our linear programming argument can be thought of as assigning types to edges based on the volume of the boxes they define, and automatically balancing the number of edges of different types by selecting the correct coefficients for the constraints corresponding to those edges.

**Steiner TC-spanners of general  $d$ -dimensional posets.** We continue the study of the number of edges in a sparsest Steiner  $k$ -TC-spanner of a poset as a function of its dimension, following [4] and [21]. Observe that the only poset of dimension 1 is the directed line  $\mathcal{H}_{n,1}$ . TC-spanners of the directed lines were discovered under many different guises. They were studied implicitly in [3, 6, 12, 13, 26] and explicitly in [9, 24]. Alon and Schieber [3] implicitly showed that, for constant  $k$ , the size of the sparsest  $k$ -TC-spanner of the directed line is  $\Theta(n \cdot \lambda_k(n))$ , where  $\lambda_k(n)$  is the  $k^{\text{th}}$ -row inverse Ackermann function.<sup>2</sup>

<sup>1</sup>For a positive integer  $m$ , we denote  $\{1, \dots, m\}$  by  $[m]$ .

<sup>2</sup>The *Ackermann function* [1] is defined by:  $A(1, j) = 2^j$ ,  $A(i + 1, 0) = A(i, 1)$ ,  $A(i + 1, j + 1) = A(i, 2^{A(i+1, j)})$ . The inverse Ackermann function is  $\alpha(n) = \min\{i : A(i, 1) \geq n\}$  and the  $i^{\text{th}}$ -row inverse is  $\lambda_i(n) = \min\{j : A(i, j) \geq n\}$ . Specifically,  $\lambda_2(n) = \Theta(\log n)$ ,  $\lambda_3(n) = \Theta(\log \log n)$  and  $\lambda_4(n) = \Theta(\log^* n)$ .

Stretch $k$	Prior bounds on $S_k(G)$	Stretch $k$	Our bounds on $S_k(G)$	
$2d - 1$	$O(n^2)$ [4]	2	$O(n \log^d n)$	$\Omega\left(n \left(\frac{\log n}{cd}\right)^d\right)$
$2d - 2 + t$ for $t \geq 2$	$O(n(\log^{d-1} n) \lambda_t(n))$ [4]		for all $d$	for a fixed $c > 0$ and all $d$
$2d + O(\log^* n)$	$O(n \log^{d-1} n)$ [4]	$\geq 3$		$\Omega(n \log^{\lceil (d-1)/k \rceil} n)$
3	$O(n \log^{d-1} n \log \log n)$ for fixed $d$ [21]			for fixed $d$

Table 1: The size of the sparsest Steiner  $k$ -TC-spanner for  $d$ -dimensional posets on  $n$  vertices for  $d \geq 2$

Table 1 compares old and new results for  $d \geq 2$ .  $S_k(G)$  denotes the number of edges in the sparsest Steiner  $k$ -TC-spanner of  $G$ . The upper bounds hold for all posets of dimension  $d$ . The lower bounds mean that there is a poset of dimension  $d$  for which every Steiner  $k$ -TC-spanner has the specified number of edges.

Atallah *et al.* construct Steiner  $k$ -TC-spanners with  $k$  proportional to  $d$ . De Santis *et al.* improved their construction for constant  $d$ . They achieve  $O(3^{d-t} n t \log^{d-1} n \log \log n)$  edges for odd stretch  $k = 2t + 1$ , where  $t \in [d]$ . In particular, setting  $t = 1$  gives  $k = 3$  and  $O(n \log^{d-1} n \log \log n)$  edges.

We present the first construction of Steiner 2-TC-spanners for  $d$ -dimensional posets. In our construction, the spanners have  $O(n \log^d n)$  edges, and the length-2 paths can be found in  $O(d)$  time. This result is stated in Theorem 2.2 (Sect. 2). Our construction takes as part of the input an explicit embedding of the poset into a  $d$ -dimensional grid. (Finding such an embedding is NP-hard [25].)

Note that the Steiner vertices used in our construction for  $d$ -dimensional posets are necessary to obtain sparse TC-spanners. Recall our example of a bipartite graph  $K_{\frac{n}{2}, \frac{n}{2}}$  for which every 2-TC-spanner required  $\Omega(n^2)$  edges.  $K_{\frac{n}{2}, \frac{n}{2}}$  is a poset of dimension 2, and thus, by the upper bound in Theorem 2.2, has a Steiner 2-TC-spanner of size  $O(n \log^2 n)$ . (As we mentioned before, for this graph there is an even better Steiner 2-TC-spanner with  $O(n)$  edges.) To see that  $K_{\frac{n}{2}, \frac{n}{2}}$  is embeddable into a  $[n] \times [n]$  grid, map each of the  $n/2$  left vertices of  $K_{\frac{n}{2}, \frac{n}{2}}$  to a distinct grid vertex in the set of incomparable vertices  $\{(i, n/2 + 1 - i) : i \in [n/2]\}$ , and similarly map each right vertex to a distinct vertex in the set  $\{(n + 1 - i, i + n/2) : i \in [n/2]\}$ . It is easy to see that this is a proper embedding.

Theorem 1.1 implies that there is an absolute constant  $c > 0$  for which our upper bound for  $k = 2$  is tight within an  $O((cd)^d)$  factor, showing that no drastic improvement in the upper bound is possible. To obtain a bound in terms of the number  $n$  of vertices and dimension  $d$ , substitute  $m^d$  with  $n$  and  $\ln m$  with  $(\ln n)/d$  in the theorem statement. This gives the following corollary.

**Corollary 1.2.** *There is an absolute constant  $c > 0$  for which for all  $d \geq 2$ , there exists a  $d$ -dimensional poset  $G$  on  $n$  vertices such that every Steiner 2-TC-spanner of  $G$  has  $\Omega\left(n \left(\frac{\log n}{cd}\right)^d\right)$  edges.*

In addition, we prove a lower bound for all constant  $k > 2$  and constant dimension  $d$ , which qualitatively matches known upper bounds. It shows that, in particular, every Steiner 3-TC-spanner has size  $\Omega(n \log n)$ , and even with significantly larger constant stretch, every Steiner TC-spanner has size  $n \log^{\Omega(d)} n$ .

**Theorem 1.3.** *For all constant  $d \geq 2$ , there exists a  $d$ -dimensional poset  $G$  on  $n$  vertices such that for all  $k \geq 3$ , every Steiner  $k$ -TC-spanner of  $G$  has  $\Omega(n \log^{\lceil (d-1)/k \rceil} n)$  edges.*

This theorem (proved in Section 4) greatly improves upon the previous  $\Omega(n \log \log n)$  bound, which follows trivially from known lower bounds for a 3-TC-Spanner of a directed line.

The lower bound on the size of a Steiner  $k$ -TC-spanner for  $k \geq 3$  is proved by the probabilistic method. We observe that using the hypergrid as an example of a poset with large Steiner  $k$ -TC-spanners for  $k > 2$  would yield a much weaker lower bound because it is known that  $\mathcal{H}_{m,d}$  has a 3-TC-spanner of size

$O((m \log \log m)^d)$  and, more generally, a  $k$ -TC-spanner of size  $O((m \cdot \lambda_k(m))^d)$ , where  $\lambda_k(m)$  is the  $k^{\text{th}}$ -row inverse Ackermann function [7]. Instead, we construct an  $n$ -element poset embedded in  $\mathcal{H}_{n,d}$  using the following randomized procedure: all poset elements differ on coordinates in dimension 1, and for each element, the remaining  $d - 1$  coordinates are chosen uniformly at random from  $[n]$ . We consider a set of partitions of the underlying hypergrid into  $d$ -dimensional boxes, and carefully count the expected number of edges in a Steiner  $k$ -TC-spanner that cross box boundaries for each partition. Then we show that each edge was counted only a small number of times, proving that the expected number of edges in a Steiner  $k$ -TC-spanner is large. We conclude that some poset attains the expected number of edges.

**Organization.** We explain applications of Steiner TC-spanners in Section 1.2. Section 2 gives basic definitions and observations. In particular, our construction of sparse Steiner 2-TC-spanners for  $d$ -dimensional posets (the proof of Theorem 2.2) is presented there. Our lower bounds are the technically hardest part of this paper. The lower bound for the hypergrid for  $k = 2$  (Theorem 1.1) is proved in Section 3. The lower bound for  $k > 2$  (Theorem 1.3) is presented in Section 4.

## 1.2 Applications

Numerous applications of TC-spanners are surveyed in [19]. We focus on two of them: property reconstruction, described in [7, 17], and key management for access control hierarchies, described in [5, 6, 21, 4, 8].

**Property Reconstruction.** Property-preserving data reconstruction was introduced by Ailon, Chazelle, Comandur and Liu [2]. In this model, a reconstruction algorithm, called a *filter*, sits between a *client* and a *dataset*. A dataset is viewed as a function  $f : \mathcal{D} \rightarrow \mathcal{R}$ . Client accesses the dataset using *queries* of the form  $x \in \mathcal{D}$  to the filter. The filter *looks up* a small number of values in the dataset and outputs  $g(x)$ , where  $g$  must satisfy some fixed *structural* property (e.g., be monotone or have a low Lipschitz constant) and differ from  $f$  as little as possible. Extending this notion, Saks and Seshadhri [20] defined *local* reconstruction. A filter is *local* if it allows for a local (or distributed) implementation: namely, if the output function  $g$  does not depend on the order of the queries.

Our results on TC-spanners are relevant to reconstruction of two properties of functions: monotonicity and having a low Lipschitz constant. Reconstruction of monotone functions was considered in [2, 20, 7]. A function  $f : [m]^d \rightarrow \mathbb{R}$  is called *monotone* if  $f(x) \leq f(y)$  for all  $(x, y) \in E(\mathcal{H}_{m,d})$ . Reconstruction of functions with low Lipschitz constant was studied in [17]. A function  $f : [m]^d \rightarrow \mathbb{R}$  has Lipschitz constant  $c$  if  $|f(x) - f(y)| \leq c \cdot |x - y|_1$ . In [7], the authors proved that the existence of a local filter for monotonicity of functions with low lookup complexity implies the existence of a sparse 2-TC-spanner of  $\mathcal{H}_{m,d}$ . In [17], an analogous connection is drawn between local reconstruction of functions with low Lipschitz constant and 2-TC-spanners. Our improvement in the lower bound on the size of 2-TC-spanners of  $\mathcal{H}_{m,d}$  directly translates into improvement by the same factor in the lower bounds on lookup complexity of local filters for these two properties.

**Key Management for Access Control Hierarchies.** Atallah *et al.* [6] used sparse Steiner TC-spanners to construct efficient key management schemes for access control hierarchies. An *access hierarchy* is a partially ordered set  $G$  of access classes. Each user is entitled to access a certain class and all classes reachable from the corresponding node in  $G$ . One approach to enforcing the access hierarchy is to use a key management scheme of the following form [5, 6, 21, 4]. Each edge  $(i, j)$  has an associated public key  $P(i, j)$ , and each node  $i$ , an associated secret key  $k_i$ . Only users with the secret key for a node have the required permissions for the associated access class. The public and secret keys are designed so that there is an efficient algorithm  $A$  which takes  $k_i$  and  $P(i, j)$  and generates  $k_j$ , but for each  $(i, j)$  in  $G$ , it is computationally hard to generate

$k_j$  without knowledge of  $k_i$ . Thus, a user can efficiently generate the required keys to access a descendant class, but not other classes. The number of runs of algorithm  $A$  needed to generate a secret key  $k_v$  from a secret key  $k_u$  is equal to  $d_G(u, v)$ . To speed this up, Atallah *et al.* [4] suggest adding edges and nodes to  $G$  to increase connectivity. To preserve the access hierarchy represented by  $G$ , the new graph  $H$  must be a Steiner TC-spanner of  $G$ . The number of edges in  $H$  corresponds to the space complexity of the scheme, while the stretch  $k$  of the spanner corresponds to the time complexity.

We note that the time to find the path from  $u$  to  $v$  is also important in this application. In our upper bounds, this time is  $O(d)$ , which for small  $d$  (e.g., constant) is likely to be much less than  $2g(n)$  or  $3g(n)$ , where  $g(n)$  is the time to run algorithm  $A$ . This is because algorithm  $A$  involves the evaluation of a cryptographic hash function, which is expensive in practice and in theory<sup>3</sup>.

## 2 Definitions and Observations

For integers  $j \geq i$ , an interval  $[i, j]$  refers to the set  $\{i, i+1, \dots, j\}$ . Unless otherwise specified, logs are always base 2, except for  $\ln$  which is the natural logarithm.

Each DAG  $G = (V, E)$  is equivalent to a poset with elements  $V$  and partial order  $\preceq$ , where  $x \preceq y$  if  $y$  is reachable from  $x$  in  $G$ . Elements  $x$  and  $y$  are *comparable* if  $x \preceq y$  or  $y \preceq x$ , and *incomparable* otherwise. We write  $x \prec y$  if  $x \preceq y$  and  $x \neq y$ . The *hypergrid*  $\mathcal{H}_{m,d}$  with dimension  $d$  and side length  $m$  was defined in the beginning of Section 1.1. Equivalently, it is the poset on elements  $[m]^d$  with the *dominance order*, defined as follows:  $x \preceq y$  for two elements  $x, y \in [m]^d$  iff  $x_i \leq y_i$  for all  $i \in [d]$ .

A mapping from a poset  $G$  to a poset  $G'$  is called an *embedding* if it respects the partial order, that is, all  $x, y \in G$  are mapped to  $x', y' \in G'$  such that  $x \preceq_G y$  iff  $x' \preceq_{G'} y'$ .

**Definition 2.1** (Poset dimension ([15])). *Let  $G$  be a poset with  $n$  elements. The dimension of  $G$  is the smallest integer  $d$  such that  $G$  can be embedded into the hypergrid  $\mathcal{H}_{n,d}$ .*

Dushnik and Miller [14] proved that for any  $m > 1$ , the hypergrid  $\mathcal{H}_{m,d}$  has dimension exactly  $d$ .

**Fact 2.1.** *Each  $d$ -dimensional poset with  $n$  elements can be embedded into a hypergrid  $\mathcal{H}_{n,d}$ , so that for all  $i \in [d]$ , the  $i$ th coordinates of images of all points are distinct.*

**Sparse Steiner 2-TC-spanners for  $d$ -dimensional posets.** We give a simple construction of sparse Steiner 2-TC-spanners for  $d$ -dimensional posets. For constant  $d$ , it matches the lower bound from Section 3 up to a constant factor. Note that the construction itself works for arbitrary, not necessary constant,  $d$ .

**Theorem 2.2.** *Every  $d$ -dimensional poset  $G$  on  $n$  elements has a Steiner 2-TC-spanner  $H$  of size  $O(n \log^d n)$  that can be constructed in time  $O(dn \log^d n)$ . Moreover, for all  $x, y \in G$ , where  $x \prec y$ , one can find a path in  $H$  from  $x$  to  $y$  of length at most 2 in time  $O(d)$ .*

*Proof.* Consider an  $n$ -element poset  $G$  embedded into the hypergrid  $\mathcal{H}_{n,d}$ , so that for all  $i \in [d]$ , the  $i$ th coordinates of images of all points are distinct. (See Fact 2.1). In this proof, assume that the hypergrid coordinates start with 0, i.e., its vertex set is  $[0, n-1]^d$ . Let  $\ell = \lceil \log n \rceil$  and  $b(t)$  be the  $\ell$ -bit binary representation of  $t$ , possibly with leading zeros. Let  $p_i(t)$  denote the  $i$ -bit prefix of  $b(t)$  followed by a single 1 and then  $\ell - i - 1$  zeros. Let  $lcp(t_1, t_2) = p_i(t_1)$ , where  $i$  is the length of the longest common prefix of  $b(t_1)$  and  $b(t_2)$ .

To construct a Steiner 2-TC-spanner  $(V_H, E_H)$  of  $G$ , we insert at most  $\ell^d$  edges into  $E_H$  per each poset element. Consider a poset element with coordinates  $x = (x_1, \dots, x_d)$  in the embedding. For each

<sup>3</sup>Any hash function which is secure against  $\text{poly}(n)$ -time adversaries requires  $g(n) \geq \text{polylog } n$  evaluation time under existing number-theoretic assumptions.

$d$ -tuple  $(i_1, \dots, i_d) \in [0, \ell - 1]^d$ , let  $p$  be a hypergrid vertex whose coordinates have binary representations  $(p_{i_1}(x_1), \dots, p_{i_d}(x_d))$ . If  $x \prec p$ , we add an edge  $(x, p)$  to  $E_H$ ; otherwise, if  $p \prec x$  we add an edge  $(p, x)$  to  $E_H$ . Note that only edges between comparable points are added to  $E_H$ .

We have that  $E_H$  contains  $O(n(\lceil \log n \rceil)^d)$  edges. If  $d = O(\log n)$ , then  $(\lceil \log n \rceil)^d \leq (\log n + 1)^d = (\log^d n)(1 + 1/\log n)^d = O(\log^d n)$ , using the well-known inequality that  $1 + x \leq e^x$ . On the other hand, if  $d = \Omega(\log n)$ , the bound of this theorem holds trivially. Hence,  $E_H$  contains  $O(n \log^d n)$  edges. It can be constructed in  $O(dn \log^d n)$  time, as described, if bit operations on coordinates can be performed in  $O(1)$  time.

For all pairs of poset elements  $x = (x_1, \dots, x_d)$  and  $y = (y_1, \dots, y_d)$ , such that  $x \prec y$ , there is an intermediate point  $z$  with coordinates whose binary representations are  $(lcp(x_1, y_1), \dots, lcp(x_d, y_d))$ . By construction, both edges  $(x, z)$  and  $(z, y)$  are in  $E_H$ . Point  $z$  can be found in  $O(d)$  time, since  $lcp(x_i, y_i)$  can be computed in  $O(1)$  time, assuming  $O(1)$  time bit operations on coordinates.  $\square$

**Equivalence of Steiner and non-Steiner TC-spanners for hypergrids.** Our lower bound on the size of 2-TC-spanners for  $d$ -dimensional posets of size  $n$  is obtained by proving a lower bound on the size of the Steiner 2-TC-spanner of  $\mathcal{H}_{m,d}$  where  $m = n^{1/d}$ . The following lemma, used in Section 4.2, implies Corollary 2.4 that shows that sparsest Steiner and non-Steiner 2-TC-spanners of  $\mathcal{H}_{m,d}$  have the same size.

**Lemma 2.3.** *Let  $G$  be a poset on elements  $V \subseteq [m]^d$  with the dominance order and  $H = (V_H, E_H)$  be a Steiner  $k$ -TC-spanner of  $G$  with minimal  $V_H$ . Then  $H$  can be embedded into  $\mathcal{H}_{m,d}$ .*

*Proof.* For each  $s \in V_H - V$ , we define  $Prev(s) = \{x \in V : x \prec s\}$ . If  $Prev(s) = \emptyset$  then  $V_H$  is not minimal because  $H$  remains a Steiner  $k$ -TC-spanner of  $G$  when  $s$  is removed. We map each Steiner vertex  $s$  to  $r(s)$ , the replacement of  $s$  in  $[m]^d$ , whose  $i$ th coordinates for all  $i \in [d]$  are  $\max_{x \in Prev(s)} x_i$ .

Consider an edge  $(x, y)$  in  $G$ . If  $x, y \in V$  our embedding does not alter that edge. If  $x \in V, y \in V_H - V$  then  $x \in Prev(y)$  and  $x \prec r(y)$  by the definition of  $r$ . If  $x, y \in V_H - V$  then  $Prev(x) \subseteq Prev(y)$  and the monotonicity of  $\max(S)$  for sets implies  $r(x) \preceq r(y)$ . Finally, if  $x \in V_H - V$  and  $y \in V$  then for each  $z \in Prev(x)$  and each  $i \in [d]$ , we have  $z_i \leq y_i$  because  $z \prec x \prec y$ , and this implies  $r(x) \preceq y$ .  $\square$

**Corollary 2.4.** *If  $\mathcal{H}_{m,d}$  has a Steiner  $k$ -TC-spanner  $H$ , it also has a  $k$ -TC-spanner with the same number of nodes and at most the same number of edges.*

### 3 Our Lower Bound for 2-TC-spanners of the Hypergrid

In this section, we prove Theorem 1.1 that gives a nearly tight lower bound on the size of (Steiner) 2-TC-spanners of the hypergrids  $\mathcal{H}_{m,d}$ . By Corollary 2.4, we only have to consider non-Steiner TC-spanners.

*Proof of Theorem 1.1.* We start by introducing a linear program for the sparsest 2-TC-spanner of an arbitrary graph. Our lower bound on the size of a 2-TC-spanner of  $\mathcal{H}_{m,d}$  is obtained by finding a feasible solution to the dual program, which, by definition, gives a lower bound on the objective function of the primal.

**Integer linear program for sparsest 2-TC-spanner.** For every graph, we can find the size of a sparsest 2-TC-spanner by solving the following  $\{0,1\}$ -linear program which is a special case of a more general program from [8] for directed  $k$ -spanners. For all vertices  $u, v$  satisfying  $u \preceq v$ , we introduce variables  $x_{uv} \in \{0,1\}$ . If  $H = (V, E_H)$  is the corresponding 2-TC-spanner,  $x_{uv} = 1$  iff  $(u, v) \in E_H$ . For all vertices  $u, v, w$  satisfying  $u \preceq w \preceq v$ , we introduce auxiliary variables  $x'_{uvw} \in \{0,1\}$ . If  $H = (V, E_H)$  is the

corresponding 2-TC-spanner,  $x'_{uvw} = 1$  if both  $(u, w)$  and  $(w, v)$  are in  $E_H$ . The  $\{0,1\}$ -linear program is as follows:

$$\begin{aligned}
& \text{minimize} && \sum_{u,v: u \preceq v} x_{uv} \\
& \text{subject to} && x_{uw} - x'_{uvw} \geq 0, x_{wv} - x'_{uvw} \geq 0 && \forall u, v, w: u \preceq w \preceq v; \\
& && \sum_{w: u \preceq w \preceq v} x'_{uvw} \geq 1 && \forall u, v: u \preceq v; \\
& && x_{uv} \in \{0, 1\} && \forall u, v: u \preceq v; \\
& && x'_{uvw} \in \{0, 1\} && \forall u, v, w: u \preceq w \preceq v.
\end{aligned}$$

The size of the sparsest 2-TC-spanner and the optimal value of the objective function of this linear program differ by  $\sum_u x_{uu} \leq m^d$ . Since we are considering asymptotic behavior of the size of the 2-TC-spanner, this difference can be ignored.

Every feasible solution of the following fractional relaxation of a dual linear program gives a lower bound on the objective function of the primal.

$$\begin{aligned}
& \text{maximize} && \sum_{u,v: u \preceq v} y_{uv} \\
& \text{subject to} && \sum_{w: v \preceq w} y'_{uvw} + \sum_{w: w \preceq u} y''_{uvw} \leq 1 && \forall u, v: u \preceq v; \tag{1} \\
& && y_{uv} - y'_{uvw} - y''_{uvw} \leq 0 && \forall u, v, w: u \preceq w \preceq v; \tag{2} \\
& && y_{uv} \geq 0 && \forall u \preceq v; \\
& && y'_{uvw} \geq 0, y''_{uvw} \geq 0 && \forall u \preceq w \preceq v.
\end{aligned}$$

**Finding a feasible solution for the dual.** The rest of the proof of the Theorem 1.1 can be broken down into the following steps:

1. We choose initial values  $\hat{y}_{uv}$  for the variables  $y_{uv}$  of the dual program and, in Lemma 3.1, give a lower bound on the resulting value of the objective function of the primal program.
2. We choose initial values  $\hat{y}'_{uvw}$  and  $\hat{y}''_{uvw}$  for variables  $y'_{uvw}$  and  $y''_{uvw}$  to ensure that (2) holds.
3. In Lemma 3.2, we give an upper bound on the left side of (1) for all  $u \preceq v$ . Our bound is a constant larger than 1 and independent of  $n$ . We obtain a feasible solution to the dual by dividing the initial variable values (and, consequently, the value of objective function) by this constant.

**Step 1.** For a vector  $x = (x_1, \dots, x_d) \in [0, m-1]^d$ , let  $V(x)$  denote  $\prod_{i \in [d]} (x_i + 1)$ . This corresponds to the number of hypergrid points inside a  $d$ -dimensional box with corners  $u$  and  $v$ , where  $v - u = x$ . To obtain the desired lower bound, we set  $\hat{y}_{uv} = \frac{1}{V(v-u)^d}$  for all  $u \preceq v$ . This gives the value of the objective function of the dual program, according to the following lemma.

**Lemma 3.1.**  $\sum_{u,v: u \preceq v} \hat{y}_{uv} > m^d (\ln m - 1)^d$

*Proof.* Substituting  $1/(V(v-u))$  for  $\hat{y}_{uv}$ , we get:

$$\begin{aligned}
\sum_{u,v: u \preceq v} \hat{y}_{uv} &= \sum_{u,v: u \preceq v} \frac{1}{V(v-u)} = \sum_{l \in [m]^d} \prod_{i \in [d]} \frac{m - l_i + 1}{l_i} = \left( \sum_{l \in [m]} \frac{m - l + 1}{l} \right)^d \\
&> ((m+1) \ln(m+1) - m)^d > m^d (\ln m - 1)^d.
\end{aligned}$$



□

**Step 2.** The values of  $\hat{y}'_{uvw}$  and  $\hat{y}''_{uvw}$  are set as follows to satisfy (2) tightly (without any slack):

$$\hat{y}'_{uvw} = \hat{y}_{uw} \frac{V(v-u)}{V(v-u) + V(w-v)}, \quad \hat{y}''_{uvw} = \hat{y}_{uw} - \hat{y}'_{uvw} = \hat{y}_{uw} \frac{V(w-v)}{V(v-u) + V(w-v)}.$$

**Step 3.** The initial values  $\hat{y}'_{uvw}$  and  $\hat{y}''_{uvw}$  do not necessarily satisfy (1). Next, we give the same upper bound on the left hand side of all constraints (1).

**Lemma 3.2.** For all  $u \preceq v$ ,  $\sum_{w: v \preceq w} \hat{y}'_{uvw} + \sum_{w: w \preceq u} \hat{y}''_{uvw} \leq (4\pi)^d$ .

*Proof.* Below we denote  $v - u$  by  $x^0 = (x_1^0, \dots, x_d^0)$ , a  $d$ -dimensional vector of ones  $(1, \dots, 1)$  as  $\vec{1}$  and  $\prod_{i \in [d]} dx_i$  by  $dx$ .

$$\begin{aligned} \sum_{w: v \preceq w} \hat{y}'_{uvw} + \sum_{w: w \preceq u} \hat{y}''_{uvw} &= \sum_{w: v \preceq w} \hat{y}_{uw} \frac{V(v-u)}{V(v-u) + V(w-v)} + \sum_{w: w \preceq u} \hat{y}_{uw} \frac{V(v-u)}{V(u-w) + V(v-u)} \\ &< 2 \sum_{x \in [0, m]^d} \frac{V(x^0)}{V(x^0 + x)(V(x^0) + V(x))} \\ &\leq 2^{2d+1} \sum_{x \in [1, m+1]^d} \frac{V(x^0)}{V(x^0 + x)(V(x^0) + V(x))} \end{aligned} \quad (3)$$

$$< 2^{2d+1} \int_{\mathbb{R}_+^d} \frac{V(x^0) dx}{V(x^0 + x)(V(x^0) + V(x))} \quad (4)$$

$$= 2^{2d+1} \int_{\mathbb{R}_+^d} \frac{V(x^0) dt}{V(t)(V(x^0) + \prod_i (t_i(x_i^0 + 1) + 1))} \quad (5)$$

$$< 2^{2d+1} \int_{\mathbb{R}_+^d} \frac{V(x^0) dt}{V(t)(V(x^0) + \prod_i t_i(x_i^0 + 1))}$$

$$= 2^{2d+1} \int_{\mathbb{R}_+^d} \frac{dt}{V(t)(\vec{1} + V(t-1))}.$$

The first equality above is obtained by plugging in values of  $\hat{y}'$  and  $\hat{y}''$  from Step 2 with appropriate indices. The first inequality is obtained by extending each sum to the whole subgrid. Here (3) holds because  $\frac{1}{V(u)} \leq \frac{2^d}{V(u+1)}$  for all  $u$ , such that  $u_i \geq 0$ . In (4), the sum can be bounded from above by the integral because the summand is monotone in all variables. To get (5), we substitute  $x$  by  $t$ , which satisfies  $x_i = t_i(x_i^0 + 1)$ . In the last inequality, we substitute  $V(x^0)$  for  $\prod_i (x_i^0 + 1)$ .

**Claim 3.3.** Let  $I_d = \int_{\mathbb{R}_+^d} \frac{dt}{V(t)(\vec{1} + V(t-1))}$ . Then  $I_d \leq \frac{\pi^d}{2}$  for all  $d$ .

Lemma 3.2 follows from Claim 3.3 whose proof is deferred to Appendix A. □

Finally, we obtain a feasible solution by dividing initial values  $\hat{y}_{uv}$ ,  $\hat{y}'_{uvw}$  and  $\hat{y}''_{uvw}$  by the upper bound  $(4\pi)^d$  from Lemma 3.2. Then Lemma 3.1 gives the desired bound on the value of the objective function.

$$\sum_{u, v: u \preceq v} \frac{\hat{y}_{uv}}{(4\pi)^d} > m^d \left( \frac{\ln m - 1}{4\pi} \right)^d$$

This completes the proof of Theorem 1.1. □

## 4 Our Lower Bound for $k$ -TC-spanners of $d$ -dimensional Posets for $k > 2$

In this section, we prove Theorem 1.3.

*Proof of Theorem 1.3.* Unlike in the previous section, the poset which attains the lower bound is constructed probabilistically, not explicitly. Let  $\mathcal{G}_d$  be a distribution on  $n$ -element posets embedded in  $\mathcal{H}_{n,d}$ , where all poset elements differ on coordinates in dimension 1, and for each such coordinate  $a \in [n]$ , an element  $p_a$  is chosen uniformly and independently from  $\{a\} \times [n]^{d-1}$ . The partial order is then given by the dominance order  $x \preceq y$  on  $\mathcal{H}_{n,d}$ .

Recall that  $S_k(G)$  denotes the size of the sparsest Steiner  $k$ -TC-spanner of poset  $G$ . The following lemma gives a lower bound on the expected size of a Steiner  $k$ -TC-spanner of a poset drawn from  $\mathcal{G}_d$ .

**Lemma 4.1.**  $\mathbb{E}_{G \leftarrow \mathcal{G}_d} [S_k(G)] = \Omega(n \log^{\lceil (d-1)/k \rceil} n)$  for all  $k \geq 3$  and constant  $d \geq 2$ .

To simplify the presentation, we first prove the special case of Lemma 4.1 for 2-dimensional posets in Section 4.1. The general case is proved in Section 4.2. Since Lemma 4.1 implies the existence of a poset  $G$ , for which every Steiner  $k$ -TC-spanner has  $\Omega(n \log^{\lceil (d-1)/k \rceil} n)$  edges, Theorem 1.3 follows.  $\square$

### 4.1 The case of $d = 2$

This section proves a special case of Lemma 4.1 for 2-dimensional posets, which illustrates many of the ideas used in the proof of the general lemma. In both proofs, we assume that  $\ell = \log n$  is an integer.

**Lemma 4.2** (Special case of Lemma 4.1).  $\mathbb{E}_{G \leftarrow \mathcal{G}_2} [S_k(G)] = \Omega(n \log n)$  for all  $k \geq 3$  and  $d = 2$ .

*Proof.* To analyze the expected number of edges in a Steiner TC-spanner, we consider  $\ell$  partitions of  $[n]^2$  into strips. We call strips *boxes* for compatibility with the case of general  $d$ .

**Definition 4.1** (Box partition). For each  $i \in [\ell]$ , we define sets of equal size that partition  $[n]$  into  $2^i$  intervals: the  $j$ th such set, for  $j \in [2^i]$ , is  $I_j^i = [(j-1)2^{\ell-i} + 1, j2^{\ell-i}]$ . Given  $i \in [\ell]$ , and  $j \in [2^i]$ , the box  $\mathbb{B}(i, j)$  is  $[n] \times I_j^i$  and the box partition  $\mathbb{BP}(i)$  is a partition of  $[n]^2$  that contains boxes  $\mathbb{B}(i, j)$  for all  $j \in [2^i]$ .

We analyze the expected number of edges that cross from boxes with an odd index  $j$  into boxes with index  $j+1$  with respect to partition  $\mathbb{BP}(i)$  for all  $i \in [\ell]$ . To do that, we identify pairs of poset elements that force such edges to appear. The pairs of their first coordinates are called *jumps* and are defined next.

**Definition 4.2** (Jumps). A jump generated by the partition  $\mathbb{BP}(i)$  is a pair  $(a, b)$  of coordinates in dimension 1, such that for some odd  $j \in [2^i]$ , the following holds:  $p_a \in \mathbb{B}(i, j)$ ,  $p_b \in \mathbb{B}(i, j+1)$ , while  $p_c \notin \mathbb{B}(i, j) \cup \mathbb{B}(i, j+1)$  for all  $c \in (a, b)$ . Let  $\mathcal{J}$  denote the set of jumps generated by all partitions  $\mathbb{BP}(i)$  for  $i \in [\ell]$ .

We use two properties of  $\mathcal{J}$ , given in Claims 4.3 and 4.4.

**Claim 4.3.** Let  $G$  be a poset, embedded into  $\mathcal{H}_{n,2}$ , and  $H = (V_H, E_H)$  be a Steiner  $k$ -TC-spanner of  $G$ . Then there exists a 1-1 mapping from  $\mathcal{J}$  to  $E_H$ .

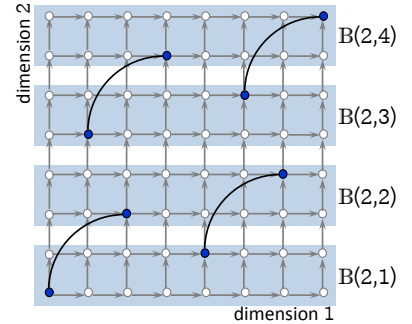


Figure 1: Box partition  $\mathbb{BP}(2)$  and jumps it generates.

*Proof.* By Lemma 2.3, we can assume that all Steiner vertices of  $H$  are embedded into  $\mathcal{H}_{n,2}$ . Given a jump  $(a, b)$ , we define  $e(a, b) \in E_H$  by following a path from  $p_a$  to  $p_b$  in  $H$ . This path is contained in  $\mathbb{B}(i, j) \cup \mathbb{B}(i, j + 1)$ , and  $e(a, b)$  is defined as the edge on that path that starts in  $\mathbb{B}(i, j)$  and ends in  $\mathbb{B}(i, j + 1)$ .

To show that  $e(a, b)$  is a 1-1 mapping, we describe an inverse mapping. To determine  $(a, b)$  from  $e(a, b) = ((u_1, u_2), (v_1, v_2))$  we find a number in  $[u_2, v_2 - 1]$ , which is divisible by the largest power of 2 and so has a form  $j2^{\ell-i}$ , from which we determine  $i$  and  $j$ . Among all jumps  $(a', b')$  defined by boxes  $\mathbb{B}(i, j), \mathbb{B}(i, j + 1)$  only one can satisfy  $a' \leq u_1 \leq v_1 \leq b'$ .  $\square$

**Claim 4.4.** *When a poset  $G$  is drawn from the distribution  $\mathcal{G}_2$ , the expected size of  $\mathcal{J}$  is at least  $n(\ell - 1)/4$ .*

*Proof.* We first find the expected number of jumps generated by the partition  $\mathbb{BP}(i)$ . We group boxes  $\mathbb{B}(i, j)$  and  $\mathbb{B}(i, j + 1)$  for odd  $j$  into box pairs. For  $u \in [n]^d$ , we define location  $\lambda_i(u)$  as such  $j$  that  $u \in \mathbb{B}(i, j)$  and parity  $\pi_i(u) = (\lambda_i(u) + 1) \bmod 2$ . Importantly, random variables  $\pi_i(p_a)$  are independent and uniform over  $\{0, 1\}$  for all  $a \in [n]$ .

We group together elements  $p_a$  that have equal values of  $\lambda_i(p_a) - \pi_i(p_a)$ , and sort elements within groups in increasing order of their first coordinate  $a$ . Observe that random variables  $\pi_i(p_a)$  within each group are uniform and independent because random variables  $\lambda_i(p_a) - \pi_i(p_a)$  and  $\pi_i(p_a)$  are independent for all  $a$ . Now, if we list  $\pi_i(p_a)$  in the sorted order for all elements in a particular group, we get a sequence of 0s and 1s. Two consecutive entries correspond to a jump iff they are 01. The last position in a group cannot correspond to the beginning of a jump. The number of positions that can correspond to the beginning of a jump in all groups is  $n$  minus the number of nonempty groups, which gives at least  $n - 2^{i-1}$ . For each such position, the probability that it starts a jump (i.e., the probability of 01) is  $1/4$ . Thus, the expected number of jumps generated by the partition  $\mathbb{BP}(i)$  is at least  $(n - 2^{i-1})/4$ .

Summing over all  $i \in [\ell]$ , we get the expected number of jumps in all partitions:  $(n\ell - \sum_{i=1}^{\ell} 2^{i-1})/4 > n(\ell - 1)/4 = \Omega(n \log n)$ .  $\square$

Claims 4.3 and 4.4 imply that, for a poset  $G$  drawn from  $\mathcal{G}_2$ , the expected number of edges in a Steiner TC-spanner  $H$  of  $G$  is  $\Omega(n \log n)$ , concluding the proof of Lemma 4.2.  $\square$

## 4.2 The case of $d > 2$

*Proof of Lemma 4.1.* Generalizing the proof for  $d = 2$ , we consider  $\ell^{d-1}$  partitions of  $[n]^d$  into boxes, where  $\ell = \log n$ . In this proof, let  $\ell' = \lfloor \ell/(d-1) \rfloor$  and  $d' = \lceil (d-1)/k \rceil$ .

**Definition 4.3** (Box partition). *Given vectors  $\vec{i} = (i_1, \dots, i_{d-1}) \in [\ell']^{d-1}$  and  $\vec{j} = (j_1, \dots, j_{d-1}) \in [2^{i_1}] \times \dots \times [2^{i_{d-1}}]$ , the box  $\mathbb{B}(\vec{i}, \vec{j})$  is  $[n] \times I_{j_1}^{i_1} \times \dots \times I_{j_{d-1}}^{i_{d-1}}$ , and the box partition  $\mathbb{BP}(\vec{i})$  is a partition of  $[n]^d$  that contains boxes  $\mathbb{B}(\vec{i}, \vec{j})$  for all eligible  $\vec{j}$ .*

To generalize the definition of the set of jumps  $\mathcal{J}$ , we denote  $(d-1)$ -dimensional vectors  $(0, \dots, 0)$  and  $(1, \dots, 1)$  by  $\vec{0}$  and  $\vec{1}$ , respectively. We say that a vector  $\vec{j}$  is *odd* if all of its coordinates are odd.

**Definition 4.4** (Jumps). *A jump generated by a box partition  $\mathbb{BP}(\vec{i})$  is a pair  $(a, b)$  of coordinates in dimension 1, such that for some vector  $\vec{i} \in [\ell']^{d-1}$  and some odd vector  $\vec{j}$ , the following holds:  $p_a \in \mathbb{B}(\vec{i}, \vec{j})$ ,  $p_b \in \mathbb{B}(\vec{i}, \vec{j} + \vec{1})$ , while  $p_c \notin \mathbb{B}(\vec{i}, \vec{j}) \cup \mathbb{B}(\vec{i}, \vec{j} + \vec{1})$  for all  $c \in (a, b)$ . The set of jumps generated by all partitions  $\mathbb{BP}(\vec{i})$  is denoted by  $\mathcal{J}$ .*

For  $u \in [n]^d$ , we define location  $\lambda_{\vec{i}}(u)$  as such  $\vec{j}$  that  $u \in \mathbb{B}(\vec{i}, \vec{j})$  and parity  $\pi_{\vec{i}}(u) = (\lambda_{\vec{i}}(u) + \vec{1}) \bmod 2$ , where mod is taken on each component of the vector.

**Claim 4.5.** *Let  $G$  be a poset, embedded into  $\mathcal{H}_{n,d}$ , and  $H = (V_H, E_H)$  be a Steiner  $k$ -TC-spanner of  $G$ . Then there exists a mapping from  $\mathcal{J}$  to  $E_H$  that maps  $O(\ell^{d-1-d'})$  jumps to one edge.*

*Proof.* By Lemma 2.3, we can assume that all Steiner vertices of  $H$  are embedded into  $\mathcal{H}_{n,d}$ .

First, we describe how to map a jump  $(a, b)$  to an edge  $e(a, b) \in E_H$ . Each jump  $(a, b)$  is generated by a box partition  $\mathbb{BP}(\vec{i})$  for some  $\vec{i}$ . We follow a path of length at most  $k$  in  $H$  from  $p_a$  to  $p_b$ , say,  $(p_a = u_0, \dots, u_k = p_b)$ , and let  $e(a, b)$  be an edge on this path that maximizes the Hamming distance between  $\pi_{\vec{i}}(u_c)$  and  $\pi_{\vec{i}}(u_{c+1})$ . Note that this distance is at least  $d'$  because  $\pi_{\vec{i}}(u_0) = \vec{0}$  and  $\pi_{\vec{i}}(u_k) = \vec{1}$ .

Now we count the jumps mapped to an edge  $e = (u, v)$ . First, we find all such jumps generated by a single box partition  $\mathbb{BP}(\vec{i})$ . They are defined by the pair of boxes  $\mathbb{B}(\vec{i}, \lambda_{\vec{i}}(u) - \pi_{\vec{i}}(u))$  and  $\mathbb{B}(\vec{i}, \lambda_{\vec{i}}(u) - \pi_{\vec{i}}(u) + \vec{1})$ . Then  $[u_1, v_1]$  must be included in one of the intervals  $[a, b]$  defined by the jumps of this pair of boxes, and those intervals are disjoint. Hence, there is at most one such jump.

It remains to count box partitions  $\mathbb{BP}(\vec{i})$  which can generate a jump mapped to a specific edge  $e$ . A necessary condition is that  $\lambda_{\vec{i}}(v) - \lambda_{\vec{i}}(u)$  is a vector in  $\{0, 1\}^{d-1}$  with at least  $d'$  1's. There are less than  $2^{d-1}$  such vectors. Consider one of these vectors, say,  $\vec{\gamma}$ . If for some  $t \in [d-1]$ ,  $\gamma_t = 1$  then  $i_t$  is uniquely determined by the largest power of 2 that divides a number in  $[u_t, v_t - 1]$ . When  $\gamma_t = 0$ , there are at most  $\ell'$  possible values of  $i_t$  because  $\vec{i} \in [\ell']^{d-1}$ . Since  $d$  is a constant, there are at most  $2^{d-1}(\ell')^{d-1-d'} = O(\ell^{d-1-d'})$  possible vectors  $\vec{\gamma}$ , such that  $\mathbb{BP}(\vec{i})$  could have generated a jump  $(a, b)$ .

Therefore,  $O(\ell^{d-1-d'})$  jumps map to the same edge of  $E_H$ .  $\square$

**Claim 4.6.** *When a poset  $G$  is drawn from the distribution  $\mathcal{G}_d$ , the expected size of  $\mathcal{J}$  is  $\Omega(\ell^{d-1}n)$ .*

*Proof.* To find the expected number of jumps generated by  $\mathbb{BP}(\vec{i})$ , we analyze the sequence  $\pi_{\vec{i}}(p_a)$ ,  $a \in [n]$ . The values in that sequence are independent and uniformly distributed over  $\{0, 1\}^{d-1}$ . First, we remove all values different from  $\vec{0}$  and  $\vec{1}$ , and obtain a sequence of expected length  $n/2^{d-2}$ . Then, in that sequence, we group together elements  $p_a$  with equal values of  $\lambda_{\vec{i}}(p_a) - \pi_{\vec{i}}(p_a)$ , and sort elements within groups in increasing order of their first coordinate  $a$ . Observe that random variables  $\pi_{\vec{i}}(p_a)$  within each group are uniform and independent because random variables  $\lambda_{\vec{i}}(p_a) - \pi_{\vec{i}}(p_a)$  and  $\pi_{\vec{i}}(p_a)$  are independent for all  $a$ . Now, if we list  $\pi_{\vec{i}}(p_a)$  in the sorted order for all elements in particular group, we get a sequence of  $\vec{0}$ s and  $\vec{1}$ s. Two consecutive entries correspond to a jump iff they are  $\vec{0}\vec{1}$ .

Let  $g(\vec{i})$  denote the number of groups, that is, the number of possible values of  $\lambda_{\vec{i}}(p_a) - \pi_{\vec{i}}(p_a)$ . Then  $g(\vec{i}) = \prod_{t=1}^{d-1} 2^{i_t-1}$ . Summing over all box partitions, we get:

$$\sum_{\vec{i} \in [\ell']^{d-1}} g(\vec{i}) = \sum_{\vec{i} \in [\ell']^{d-1}} \prod_{t=1}^{d-1} 2^{i_t-1} = \left( \sum_{t=1}^{\ell'} 2^{t-1} \right)^{d-1} < 2^{\ell'(d-1)} \leq 2^\ell = n.$$

On every position in the reordered sequence that is not the final position in its group, the expected number of jumps started is  $1/4$ , so the expected number of jumps is at least  $(n/2^{d-2} - g(\vec{i}))/4 = n/2^d - g(\vec{i})/4$ . Therefore, the expected number of jumps generated by all box partitions is at least

$$(\ell')^{d-1}n/2^d - \frac{1}{4} \sum_{\vec{i} \in [\ell']^{d-1}} g(\vec{i}) \geq (\ell')^{d-1}n/2^d - n/4 = \Omega(\ell^{d-1}n).$$

The last equality holds because  $d$  is constant.  $\square$

Claim 4.6 gives a lower bound of  $\Omega(\ell^{d-1}n)$  on the expected number of jumps in a poset  $G$ . The mapping from Claim 4.5 takes  $O(\ell^{d-1-d'})$  of these jumps to one edge. Thus, the expected number of edges in a Steiner TC-spanner  $H$  of  $G$  is  $\Omega(n\ell^{d'}) = \Omega(n \log^{\lceil (d-1)/k \rceil} n)$ . This concludes the proof of Lemma 4.1.  $\square$

## References

- [1] W. Ackermann. Zum Hilbertschen aufbau der reellen zahlen. *Math. Ann.*, 99:118–133, 1928.
- [2] N. Ailon, B. Chazelle, S. Comandur, and D. Liu. Property-preserving data reconstruction. *Algorithmica*, 51(2):160–182, 2008.
- [3] N. Alon and B. Schieber. Optimal preprocessing for answering on-line product queries. Technical Report 71/87, Tel-Aviv University, 1987.
- [4] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken. Dynamic and efficient key management for access hierarchies. *ACM Trans. Inf. Syst. Secur.*, 12(3):1–43, 2009.
- [5] M. J. Atallah, M. Blanton, and K. B. Frikken. Key management for non-tree access hierarchies. In *SACMAT*, pages 11–18, 2006.
- [6] M. J. Atallah, K. B. Frikken, N. Fazio, and M. Blanton. Dynamic and efficient key management for access hierarchies. In *ACM Conference on Computer and Communications Security*, pages 190–202, 2005.
- [7] A. Bhattacharyya, E. Grigorescu, M. Jha, K. Jung, S. Raskhodnikova, and D. Woodruff. Lower bounds for local monotonicity reconstruction from transitive-closure spanners. In *Proceedings of the 14th RANDOM*, pages 448–461, 2010.
- [8] A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. P. Woodruff. Transitive-closure spanners. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 932–941, 2009.
- [9] H. L. Bodlaender, G. Tel, and N. Santoro. Trade-offs in non-reversing diameter. *Nordic J. of Computing*, 1(1):111–134, 1994.
- [10] A. K. Chandra, S. Fortune, and R. J. Lipton. Lower bounds for constant depth circuits for prefix problems. In *Proc. 10th Annual International Conference on Automata, Languages, and Programming*, pages 109–117, 1983.
- [11] A. K. Chandra, S. Fortune, and R. J. Lipton. Unbounded fan-in circuits and associative functions. In *Proc. 15th Annual ACM Symposium on the Theory of Computing*, pages 52–60, 1983.
- [12] B. Chazelle. Computing on a free tree via complexity-preserving mappings. *Algorithmica*, 2:337–361, 1987.
- [13] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky. Improved testing algorithms for monotonicity. In *RANDOM*, pages 97–108, 1999.
- [14] B. Dushnik and E. Miller. Concerning similarity transformations of linearly ordered sets. *Bulletin Amer. Math. Soc.*, 46:322–326, 1940.
- [15] B. Dushnik and E. W. Miller. Partially ordered sets. *American Journal of Mathematics*, 63:600–610, 1941.
- [16] W. Hesse. Directed graphs requiring large numbers of shortcuts. In *SODA*, pages 665–669, 2003.
- [17] M. Jha and S. Raskhodnikova. Testing and reconstruction of Lipschitz functions with applications to data privacy. Manuscript, 2010.
- [18] D. Peleg and A. A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.
- [19] S. Raskhodnikova. Transitive-closure spanners: a survey. In O. Goldreich, editor, *Property Testing*, volume 6390 of *LNCS State-of-the-Art Surveys*, pages 167–196. Springer, Heidelberg, 2010.
- [20] M. E. Saks and C. Seshadhri. Parallel monotonicity reconstruction. In *Proceedings of the 19th Annual Symposium on Discrete Algorithms (SODA)*, pages 962–971, 2008.
- [21] A. D. Santis, A. L. Ferrara, and B. Masucci. Efficient provably-secure hierarchical key assignment schemes. In *MFCS*, pages 371–382, 2007.
- [22] M. Thorup. On shortcutting digraphs. In *WG*, pages 205–211, 1992.
- [23] M. Thorup. Shortcutting planar digraphs. *Combinatorics, Probability & Computing*, 4:287–315, 1995.

- [24] M. Thorup. Parallel shortcutting of rooted trees. *J. Algorithms*, 23(1):139–159, 1997.
- [25] M. Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on Matrix Analysis and Applications*, 3(3):351–358, 1982. <http://dx.doi.org/10.1137/0603036>.
- [26] A. C.-C. Yao. Space-time tradeoff for answering range queries (extended abstract). In *STOC*, pages 128–136, 1982.

## A Missing Proofs from Section 3: Estimating the Integral $I_d$

*Proof of Claim 3.3.* To bound the integral  $I_d$ , we first make a substitution  $x_i = \frac{1-t_i}{1+t_i}$ :

$$I_d = \int_{[-1\dots 1]^d} \frac{dx}{\prod_{1 \leq i \leq d} (1 + x_i) + \prod_{1 \leq i \leq d} (1 - x_i)}.$$

Then we bound the denominator using the inequality  $a + b \geq 2\sqrt{ab}$  and get

$$I_d \leq \int_{[-1\dots 1]^d} \frac{dx}{2 \sqrt{\prod_{1 \leq i \leq d} (1 + x_i) \times \prod_{1 \leq i \leq d} (1 - x_i)}} = \frac{J^d}{2},$$

where  $J$  denotes the following integral:

$$J = \int_{-1}^1 \frac{dx}{\sqrt{1 - x^2}} = \pi.$$

Therefore,  $I_d \leq \frac{\pi^d}{2}$ , as claimed. □